# 移动代理服务器 MAS

# 短信 API2.2

# 开发手册

# （JAVA）

**V2.2**

深圳市嘉讯软件有限公司

二〇〇七年十一月

# 目　录

1 引言.....1
1.1 编写目的.....1
1.2 背景说明.....1
1.3 术语定义.....1
2 概述.....3
2.1 接口功能.....3
2.2 接口组成.....4
2.3 短信收发机制.....4
2.3.1 MT 短信与其回执、回复的对应.....4
2.3.2 MO 短信的分流.....5
3 接口介绍.....6
3.1 接口管理模块.....6
3.1.1 主页面.....6
3.1.2 新建页面.....7
3.1.3 编辑页面.....9
3.1.4 队列管理页面.....9
3.2 数据库 DB 接口.....10
3.2.1 发送队列表.....11
3.2.2 接收队列表.....13
3.2.3 回执队列表.....13
3.3 Java API.....13
3.3.1 类 APIClient.....14
3.3.2 类 MOItem.....28
3.3.3 类 RPTItem.....29
4 使用指导.....29
4.1 开发步骤.....29
4.2 调试方法.....30
4.3 运行设置与监控.....31
5 附录.....32
5.1 附录一：API 返回值列表.....32
5.2 附录二：回执编码列表.....33

# 1 引言

## 1.1 编写目的

移动代理服务器 **MAS** 提供了强大的短信接入机制。本手册用于指导客户利用移动代理服务器 **MAS** 提供的短信接口进行其他需要短信功能的新系统开发或原有系统的短信功能扩展。

本手册的阅读对象为需要采用移动代理服务器 **MAS** 的短信接口完成客户开发的设计人员与编码人员。

## 1.2 背景说明

本手册适用于产品“移动代理服务器 **MAS**” **V1.1 + MAS1.1 P08D071121** 通用补丁。

## 1.3 术语定义

| | |
|---|---|
| **移动代理服务器 MAS：** | 全称为“移动代理服务器 **MAS**”，属深圳嘉讯软件公司开发的一套产品，本手册中有时也直接简称为“移动代理服务器”或“**MAS**”。 |
| **MT 短信：** | 通过短信接口发往手机终端的短信，也指从移动代理服务器的某一功能直接发往手机终端的短信。 |
| **MO 短信：** | 从手机终端发往移动代理服务器的短信，这些短信根据特定的规则有的直接转交到移动代理服务器的某一具体功能，有的转交到某一短信接口的接收队列。 |
| **短信回执：** | 有时也称“状态报告”，对于 **MT** 短信，手机终端收到后会返回短信回执给移动代理服务器。对于短信接口，在创建接口时，回执选择为支持回执，则通过发送队列发送出去的所有 **MT** 短信随后都会在回执队列中收到对应的回执。回执包含 **MT** 短信是否成功提交到特定手机终端的信息。 |

| | |
|---|---|
| **短信回复：** | 对于 **MT** 短信，手机终端收到后可以使用短信回复功能回复一条短信，对于通过发送队列发送的 **MT** 短信，其回复短信随后都可以从接收队列中获得。如果接口设置为支持短号，则不能直接回复接口所发出的短信。 |
| **接口编码：** | 某套短信接口的唯一标识，在创建接口时设置，随后不允许修改。 |
| **短信子码：** | 也称“短信子号”，每套短信接口都拥有一个特定数字，这个特定数字由短信收发机制所使用，其具体作用请参照章节 **2.3.1** |
| **手机指令：** | 有时也称“命令字”或“指令”，每套短信接口可以设置一个 手机指令，由字母或数字组成的，用户可以使用此指令主动发送 **MO** 短信到 **MAS**，其作用请参照章节 **2.3.2** |

# 2 概述

移动代理服务器 MAS 不仅为各种类型的集团客户直接提供了丰富的短信业务应用功能，而且提供强大的短信接入机制。

此机制允许客户再接入任何其他系统以开发此系统的短信应用功能，从而使拥有移动代理服务器 MAS 的客户相当于拥有了一个属于自己的“袖珍型短信行业网关”

## 2.1 接口功能

移动代理服务器 MAS 的短信接入机制所提供的功能是很强大的：

- 支持 MT 短信的发送、MO 短信的接收。
- 所发送的 MT 短信可包括多达 2000 个汉字。
- 支持 MT 短信的回执确认。
- 支持 MT 短信的回复会话。
- 支持自定义 MT 短信在手机上显示的号码。
- 支持定时发送 MT 短信。
- 可提供多套短信接口，从而支持多套不同系统的接入。
- 支持通过分配给接口的短信子码来接收 MO 短信。
- 支持通过接口设置的手机指令来接收 MO 短信。
- 对通过接口发送的短信可以进行短信内容过滤。
- 对通过接口发送的短信可以进行黑白名单的控制。
- 所有接口的收发短信都有话单记录
- 可以出丰富的接口短信话单报表。
- 提供接入的多种语言开发包。

- 提供不受任何具体开发语言约束的数据库接入。
- 提供开发用的调试环境。
- 提供短信收发的监控功能。
- 支持 WAP Push 短信。

## 2.2 接口组成

移动代理服务器 MAS 的短信接口包括如下部分：

| 组成部分 | 功能说明 |
|---|---|
| 接口管理模块 | 用于短信接口的创建、设置、监控；用于开发时的辅助调试。 |
| 各种语言包 | 用于各种语言的 API 调用，支持目前主流的语言，如：C++，DotNet，Java，ASP，VB，PowerBuilder，Delphi 等。 |
| 数据库接口 | 用于其他系统直接通过访问数据库来进行短信的收发。 |

## 2.3 短信收发机制

### 2.3.1 MT 短信与其回执、回复的对应

外部系统利用移动代理服务器 MAS 的短信接口进行 MT 短信的发送时，可以为每条 MT 短信提供指定一个短信编号（也称短信 ID），那么随后这条短信的回执就会带上同样的短信编号。这样外部系统通过发送 MT 短信时所用的短信编号就可以辨识出随后针对这条短信的回执。

短信编号可以是 1 到 99999999 之间的任何一个整数。如果外部系统需要随后能辨别出每条 MT 短信的回执功能，那么外部系统必须确保在一定的时间段内所发送的每条 MT 短信都有唯一的短信编号。

短信下发时还可以指定一个在手机上显示的尾号（也称 **SRCID**），回复短信时，就是通过 **srcid** 识别对应短信的。如果不指定 **srcid**，则使用短信编号作为手机尾号。例如：**sp** 特服号为：**106910050**，短信子码为：**16**，**srcid** 指定为：**123**，则发送到手机上显示的号码为：**10691005016123.**用户回复此条短信，则外部系统可以正确收到此条短信的回复。

对于某些 **MT** 短信，外部系统可能不需要辨识其回执、回复。对于这类短信，在发送时都可以指定其短信编号为 **0**。

### 2.3.2 MO 短信的分流

移动代理服务器 **MAS** 可以支持几十套短信接口，那么移动代理服务器 **MAS** 怎样确保手机用户主动发起的 **MO** 短信能准确的达到某套短信接口呢？这有如下两种方法：

**短信子码分流**。在创建一套短信接口时，移动代理服务器会要求用户为此接口分配一个短信子码，随后 **MO** 短信就可以通过短信子码的辨识而到达此接口。举例如下：

假设发往此信移动代理服务器所用的目的号码为 **106910050**，而某 **1** 套短信接口所分配的短信子码为 **5**，则手机用户若要发送短信到此接口，只要发往目的号码 **1069100505** 则可。

**手机指令分流**。对于每套短信接口，移动代理服务器允许客户为此短信接口指定手机指令，那么随后以此手机指令提交的短信，即使目的号码不带短信子码，此短信也可到达此短信接口。举例如下：

假设发往此移动代理服务器所用的目的号码为 **106910050**，短信分隔符是“**#**”，而某一套短信接口设置有手机指令为 **BANK**，则手机用户发往 **106910050** 的所有短信，只要短信内容为 **BANK#……**，则 **BANK#**后面的短信内容都会到达此接口。

# 3 接口介绍

## 3.1 接口管理模块

登录移动代理服务器后，只要用户权限允许，用户可以在系统管理菜单下访问“接口管理”模块。利用此功能模块，可以对接口进行图形化创建、维护、设置、监控、或进行客户开发的调试辅助。

如果用户无法访问“系统管理 —〉接口管理”，应联系移动代理服务器的系统管理员授权。

### 3.1.1 主页面

接口管理的主页面如下：

接口管理包括短信接口和彩信接口管理。

短信接口管理的主页面主要包括 **4** 个按钮与一个列表。所创建的每个接口在列表上有 **1** 条记录，通过此列表可以查阅每个接口的基本信息。

列表上方各按钮功能如下：

- **新建**：用于创建一套新的短信接口，
- **编辑**：用于编辑已存在短信接口的某些属性
- **删除**：删除一套已存在的短信接口。

- **队列管理**：可以查阅正在发送队列、接收队列、回执队列中排队的短信记录，同时用于清除一些非法的短信记录。配合暂停功能（调试状态），可用于外系统短信接入开发中的调试。

### 3.1.2 新建页面

通过主页面的“新建”按钮创建接口时，就会弹出如下新建页面：

通过如上页面就可以创建一个接口。如上页面的各输入元素如下：

- **接口编码**：接口的编码标识符，任意英文字母及数字组成，不能重复。
- **接口名称**：接口的描述性名称，便于开发用户记忆方便。
- **接口类型**：包括 DB 类型、API 类型，WEB Service 和 Socket 接口四种类型。DB 接口：表示开发用户可以通过任何途径（JDBC，ODBC，其它 API 等）连接上 MAS 的数据库，即可进行短信的收发。API 类型，是提供各种语言的 API 接口，方便用户进行开发。WEB Service 类型提供可以以 Web Service 的方式进行短信收发。Socket 类型提供可

以以 **Socket** 的方式进行短信收发。后两种接口的开发手册不在此文档中描述，另有文档专门对其说明。

- **业务类型**：接口所属的业务应用类型，设置时应该设置业务已开通短信能力的业务类型。否则接口的短信不能发送出去。
- **业务代码**：接口发送短信时使用的业务代码。业务代码在“系统管理->业务设置->业务代码维护”中管理。
- **短信子码**：也称“短信子号”，其值是由“系统管理->业务设置->短信子号维护”中设置的。已经使用过的子码不会在下拉列表中出现。
- **手机指令**：可以为接口设置手机指令，当主动上发短信到接口时可以使用此指令发送，也可以不设置。
- **短号支持**：包括启用，关闭；启用表示发送出去的短信 **srcID** 不带流水号，这样的短信就不能直接通过手机回复到接口。
- **回执要求**：包括支持回执，不支持回执，后者在 **DB** 接口中不提供回执队列表，在 **API** 接口中回执获取函数返回永远为 **null.**
- **登录用户**：对应 **DB** 接口建立数据库连接的用户名，或 **API** 接口 **init** 函数的 **loginName** 参数。
- **登录密码**：对应 **DB** 接口建立数据库连接的用户密码，或 **API** 接口 **init** 函数的 **loginPWD** 参数。
- **接口状态**：包括运行状态、调试状态和暂停（失效）状态三种类型。当在接口连接开发中时，应设置接口为调试状态，这时候队列管理页面可作为调试时的模拟网关，此时短信接口是还没有接入真实的短信网关的。当开发完投入运行时，应设置接口为运行状态，这样短信接口就接入了真实的短信网关。当接口处于暂停（失效）状态时，接口不起作用，不发送或接收短信。
- **失效日期**：**API** 接口的失效日期。当接口状态为：运行或者调试状态时，设置的失效日期必须大于当前日期；当接口到达失效日期后，接口状态自动设置为暂停（失效）状态。

### 3.1.3 编辑页面

通过主页面的“编辑”按钮编辑某接口时，就会弹出如下编辑页面：

通过如上页面就可以修改一个接口。页面上的各个元素的含义与新增接口页面是一样的，但有如下几个元素不能修改：

- **接口编码：**编码不能修改。
- **登录用户：**登录用户名称不能修改。

### 3.1.4 队列管理页面

通过主页面的“队列管理”按钮，就可以进入如下队列管理页面：

- **发送队列**：利用此页面可查看或清除尚在发送队列表的某些记录。
- **接收队列**：利用此页面可查看或清除尚在接收队列表的某些记录。
- **回执队列**：利用此页面可查看或清除尚在回执队列表的某些记录。

在调试状态时，如上 **Tab** 子页都提供了“删除”“添加”按钮，这样利用其添加、删除记录的功能，就可以模拟短信网关接收上行的 **MT** 短信，并下行 **MO** 短信、**MT** 短信回执。

在运行状态时，如上 **Tab** 子页上不会出现“删除”“添加”按钮。用户仅能通过队列管理页面查看当前 **3** 个队列的排队情况。

## 3.2 数据库 DB 接口

**DB** 接口存在于一个库名为“**mas**”的 **MySQL** 库内，**MySQL** 数据库的版本为“**4.0.20**”，用户可通过 **JDBC**，**ODBC**，**DAO** 等任何 **MySQL** 可支持的方式访问。建立数据库连接时，所用的 **IP** 地址应为移动代理服务器的 **IP** 地址，登录名、登录密码为接口创建时的接口登录名、接口登录密码，数据库名一般情况下使用“**mas**”即可。

**JDBC**，**ODBC**，**DAO** 驱动可以从 **mysql** 的官方网站上下载。不同的开发语言连接 **DB** 使用的连接驱动程序亦可从网上查到。

**DB** 接口由如下 **3** 个表组成，**API MT** 队列表用于发送短信，**API MO** 队列表用于接收短信，**API RPT** 队列表用于接收回执。如果创建接口时，不要求支持回执，则 **API RPT** 队列表不存在。

| API MT队列 | | |
|---|---|---|
| 自动编号 | bigint unsigned | <pk> |
| 短信ID | decimal(10,0) unsigned | |
| 终端源地址 | decimal(10,0) unsigned | |
| 手机号码集 | longtext | |
| 短信内容 | text | |
| 是否WAP Push | tinyint | |
| URL | varchar(110) | |
| 发送时间 | datetime | |
| 短信类型 | tinyint(1) | |
| 短信编码格式 | int(11) | |
| TP_PID | tinyint(1) | |
| TP_UDHI | tinyint(1) | |
| 被计费用户的号码 | varchar(10) | |
| 资费类别 | varchar(10) | |
| 资费代码 | varchar(10) | |
| 计费用户类型 | tinyint(11) | |

| API 回执队列 | | |
|---|---|---|
| 自动编号 | bigint unsigned | <pk> |
| 短信ID | decimal(10,0) unsigned | |
| 手机号码 | varchar(15) | |
| 回执编码 | smallint | |
| 回执内容 | varchar(255) | |
| 回执时间 | datetime | |

| API MO队列 | | |
|---|---|---|
| 自动编号 | bigint unsigned | <pk> |
| 短信ID | decimal(10,0) unsigned | |
| 手机号码 | varchar(15) | |
| 短信内容 | varchar(160) | |
| MO时间 | datetime | |
| MO短信编码格式 | int(11) | |

## 3.2.1 发送队列表

此表用于发送短信，表名为 **api_mt_<xxx>**，其中**<xxx>**为接口编码。表结构如下：

| 字段名 | 数据类型 | 长度 | 键 | 允许空 | 说明 |
|---|---|---|---|---|---|
| AUTO_SN | Unsigned bigint | | PK | N | 自增量 |
| SM_ID | Unsigned Decimal(8,0) | | | N | 短信 ID，0 到 99999999 之间的任何一个数值。 缺省值 0。<br>当 SM_ID 为 0 时，表示这类短信不需要辨别其回执、回复。 |
| SRC_ID | Unsigned Decimal(8,0) | | | N | 手机上显示尾号（终端源地址），0 到 99999999 之间的任何一个数值。 |
| MOBILES | text | | | N | 用英文逗号或中文逗号隔开的手机号码集 |

| 字段名 | 数据类型 | 长度 | 键 | 允许空 | 说明 |
|---|---|---|---|---|---|
| CONTENT | text | | | N | 短信内容。<br>普通短信：最大支持 2000 个汉字的短信，超过 2000 时会被截断。<br><br>PDU 短信：最大支持 140 个字节的短信，超过 140 个字节，发送时会被截断。 |
| IS_WAP | tinyint | 1 | | N | 是否 WAP Push 短信的标识，1：表示是，0：表示否，默认为 0 |
| URL | varchar | 110 | | N | WAP Push 短信的 URL，默认为””，如果数据与 IS_WAP 值不符合时以此值为准。 |
| SEND_TIME | datetime | | | Y | 短信定时发送时间，格式为：yyyy-MM-dd HH:mm:ss，为 null 时立即发送。 |
| SM_TYPE | tinyint | 1 | | N | 短信类型，默认为 0：普通短信，PDU 短信：值为 2。 |
| MSG_FMT | int | 11 | | N | 短信编码格式(最终通过 cmpp2.1 协议发送至网关的短信格式，具体类型请参见 cmpp2.1 或 cmpp3.0 协议文档;)<br>默认为 0,只有当 SM_TYPE 值为 2 时，该字段有效。 |
| TP_PID | tinyint | 1 | | N | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| TP_UDHI | tinyint | 1 | | N | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| FEE_TERMINAL_ID | varchar | 10 | | Y | 被计费用户的号码，默认为空。 |
| FEE_TYPE | varchar | 10 | | Y | 资费类别，默认为空。 |
| FEE_CODE | varchar | 10 | | Y | 资费代码，默认为空。 |
| FEE_USER_TYPE | int | 11 | | N | 计费用户类型，默认为 0。<br>当值为-1 时，上面三个计费参数（FEE_TERMINAL_ID、FEE_TYPE、FEE_CODE）无效，MAS 自动取通讯设置中的计费参数值。 |

外部系统可对此表进行 **SELECT**、**INSERT** 操作。外部系统通过插入 **1** 条记录来进行一次 **MT** 短信发送。

当为 **WAP Push** 短信时，**URL** 长度不能超过 **85** 字节，**URL** 和 **CONTENT** 长度之和不能超过 **120** 个字节。

### 3.2.2 接收队列表

此表用于接收短信，表名为 **api_mo_<xxx>**，其中**<xxx>**为接口编码。表结构如下：

| 字段名 | 数据类型 | 长度 | 键 | 允许空 | 说明 |
|---|---|---|---|---|---|
| AUTO_SN | Unsigned bigint | | PK | N | 记录序列号 |
| SM_ID | Unsigned Decimal(8,0) | | | N | MO 短信的短信 ID ，对应发送队列表（MT 表）的 SRC_ID 字段。 |
| MOBILE | varchar | 15 | | N | 发送短信的手机号码 |
| CONTENT | text | | | | 短信内容 |
| MO_TIME | datetime | | | N | MO 短信的时间 |
| MSG_FMT | int | 11 | | N | MO 短信的编码格式 |

外部系统可对此表进行 **SELECT**、**DELETE** 操作。外部系统通过 **SELECT** 操作来取出所有已收到的 **MO** 短信，取出后的记录通过 **DELETE** 操作来删除掉。

### 3.2.3 回执队列表

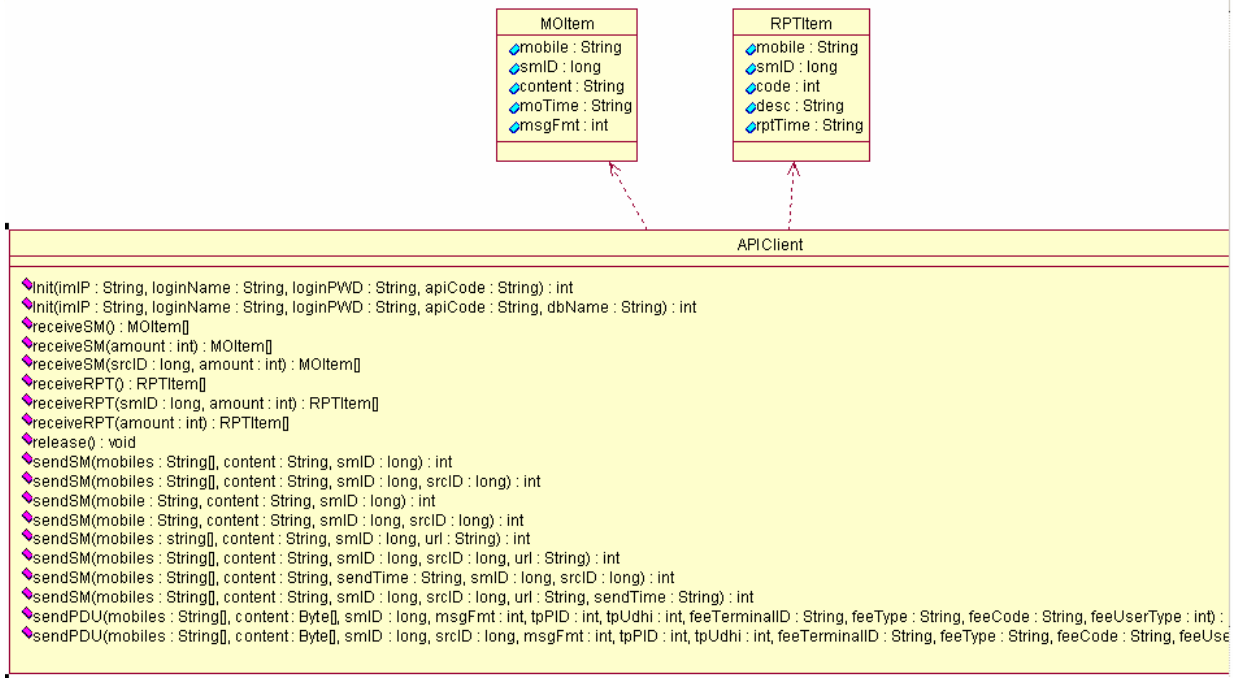
此表用于接收回执，表名为 **api_rpt_<xxx>**，其中**<xxx>**为接口编码。表结构如下：

| 字段名 | 数据类型 | 长度 | 键 | 允许空 | 说明 |
|---|---|---|---|---|---|
| AUTO_SN | Unsigned bigint | | PK | N | 记录序列号 |
| SM_ID | Unsigned Decimal(8,0) | | | N | MT 短信的短信 ID，对应发送队列表（MT 表）的 SM_ID 字段。 |
| MOBILE | varchar | 15 | | N | MT 所发送的手机号码 |
| RPT_CODE | smallint | | | N | 回执编码 |
| RPT_DESC | varchar | 100 | | | 回执的说明文字 |
| RPT_TIME | datetime | | | N | 回执的时间 |

外部系统可对此表进行 **SELECT**、**DELETE** 操作。外部系统通过 **SELECT** 操作来取出所有已返回的短信回执，取出后的记录通过 **DELETE** 操作来删除掉。

## 3.3 Java API

**JAVA** 开发包的文件名为 **ImApi.jar**，是基于 **JDK1.4.2** 开发的，可在发布光盘中找到。如下类图，开发包包括 **APIClient**、**MOItem**、**RPTItem** 等 **3** 个类，**3** 个类的包路径都为：**com.jasson.im.api**。


MOItem
mobile : String
smID : long
content : String
moTime : String
msgFmt : int
RPTItem
mobile : String
smID : long
code : int
desc : String
rptTime : String
APIClient
Init(imIP : String, loginName : String, loginPWD : String, apiCode : String) : int
Init(imIP : String, loginName : String, loginPWD : String, apiCode : String, dbName : String) : int
receiveSM() : MOItem[]
receiveSM(amount : int) : MOItem[]
receiveSM(srcID : long, amount : int) : MOItem[]
receiveRPT() : RPTItem[]
receiveRPT(smID : long, amount : int) : RPTItem[]
receiveRPT(amount : int) : RPTItem[]
release() : void
sendSM(mobiles : String[], content : String, smID : long) : int
sendSM(mobiles : String[], content : String, smID : long, srcID : long) : int
sendSM(mobile : String, content : String, smID : long) : int
sendSM(mobile : String, content : String, smID : long, srcID : long) : int
sendSM(mobiles : string[], content : String, smID : long, url : String) : int
sendSM(mobiles : String[], content : String, smID : long, srcID : long, url : String) : int
sendSM(mobiles : String[], content : String, sendTime : String, smID : long, srcID : long) : int
sendSM(mobiles : String[], content : String, smID : long, srcID : long, url : String, sendTime : String) : int
sendPDU(mobiles : String[], content : Byte[], smID : long, msgFmt : int, tpPID : int, tpUdhi : int, feeTerminalID : String, feeType : String, feeCode : String, feeUserType : int) :
sendPDU(mobiles : String[], content : Byte[], smID : long, srcID : long, msgFmt : int, tpPID : int, tpUdhi : int, feeTerminalID : String, feeType : String, feeCode : String, feeUse

## 3.3.1 类 APIClient

短信发送、短信接收、回执接收类。

**属性**

public final static int IMAPI_SUCC = 0; // 操作成功

public final static int IMAPI_CONN_ERR = -1; // 连接数据库出错

public final static int IMAPI_CONN_CLOSE_ERR = -2; // 数据库关闭失败

public final static int IMAPI_INS_ERR = -3; // 数据库插入错误

public final static int IMAPI_DEL_ERR = -4; // 数据库删除错误

public final static int IMAPI_QUERY_ERR = -5; // 数据库查询错误

public final static int IMAPI_DATA_ERR = -6; // 参数错误

public final static int IMAPI_API_ERR = -7; // API 编码非法

public final static int IMAPI_DATA_TOOLONG = -8; // 参数超长

public final static int IMAPI_INIT_ERR = -9; // 没有初始化或初始化失败

public final static int IMAPI_IFSTATUS_INVALID = -10; // API 接口处于暂停（失效）状态

public final static int IMAPI_GATEWAY_CONN_ERR = -11; // 短信网关未连接

public final static int SM_TYPE_NORMAL = 0; // 短信类型：普通短信

public final static int SM_TYPE_PDU = 2; // 短信类型：PDU 短信

**方法**

#### 3.3.1.1 init(imIP , loginName, loginPWD, apiCode)

本方法用于初始化。包括建立到移动代理服务器的连接，分配所属资源。此方法是为兼容原来 **API2.0** 版本而保留的方法，此方法没有数据库名，默认情况下为“**im**”，所以在移动代理服务器中应该使用下面一个带 **dbName** 参数的 **init** 方法，并且 **dbName** 为“**mas**”。

原型： public int init (String imIP , String loginName, String loginPWD, String apiCode)

参数：

| | |
|---|---|
| imIP | 移动代理服务器的 IP 地址 |
| loginName | 接口创建时的接口登录名 |
| loginPWD | 接口创建时的接口登录密码 |
| apiCode | 接口编码 |

返回值： Int. IMAPI_SUCC，IMAPI_CONN_ERR, IMAPI_API_ERR

#### 3.3.1.2 init(imIP , loginName, loginPWD, apiCode, dbName)

本方法用于初始化。包括建立到移动代理服务器的连接，分配所属资源。

原型： public int init (String imIP , String loginName, String loginPWD, String apiCode, String dbName)

参数：

| | |
|---|---|
| imIP | 移动代理服务器的 IP 地址 |
| loginName | 接口创建时的接口登录名 |
| loginPWD | 接口创建时的接口登录密码 |
| apiCode | 接口编码 |
| dbName | 数据库名称 |

返回值： Int. IMAPI_SUCC，IMAPI_CONN_ERR, IMAPI_API_ERR

#### 3.3.1.3 sendSM(mobile, content , smID)

发送一条 MT 短信到某个手机号码。

原型： public int sendSM(String mobile, String content , long smID)

参数：

| | |
|---|---|
| mobile | 短信发送的目的手机号码 |
| content | 普通短信内容，最大支持 2000 个汉字的短信，超过 2000 时会被截断。 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |

返回值： Int. IMAPI_SUCC , IMAPI_INIT_ERR IMAPI_CONN_ERR , IMAPI_DATA_ERR , IMAPI_DATA_TOOLONG ,

IMAPI_IFSTATUS_INVALID ，
IMAPI_GATEWAY_CONN_ERR

### 3.3.1.4 sendSM(mobiles, content , smID)

发送一条 MT 短信到一个手机号码集。

**原型：** public int sendSM(String[] mobiles, String content , long smID)

**参数：**

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | 普通短信内容，最大支持 2000 个汉字的短信，超过 2000 时会被截断。 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |

**返回值：** Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

### 3.3.1.5 sendSM(mobiles, content , smID , srcID)

发送一条 MT 短信到一个手机号码集。

**原型：** public int sendSM(String[] mobiles, String content , long smID , long srcID )

**参数：**

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | 普通短信内容，最大支持 2000 个汉字的短信，超过 2000 时会被截断。 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执 |
| srcID | 终端源地址，0 到 99999999 中的某一整数。用作源地址显示在终端上，确保唯一后可以用来找到对应的回复 |

**返回值：** Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

#### 3.3.1.6 sendSM(mobile, content , smID, URL)

发送一条 **WAP Push** 短信到某个手机号码。

**原型：** public int sendSM(String mobile, String content , long smID , String url)

**参数：**

| | |
|---|---|
| mobile | 短信发送的目的手机号码 |
| content | WAP Push 短信的标题 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |
| url | WAP Push 短信的 URL，最大长度不能超过 85 字节，为“”或 null 时则发送常规短信。当 url 不为空时， |

content 为 Push 短信的标题，此时 content 和 url 的长度之和不能超过 120 字节。

返回值： Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

### 3.3.1.7 sendSM(mobiles, content , smID, URL)

发送一条 **WAP Push** 短信到手机号码集。

原型： public int sendSM(String[] mobiles, String content , long smID , String url)

参数：

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | WAP Push 短信的标题 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |
| url | WAP Push 短信的 URL，为“”或 null 时则发送常规短信，此时 content 为短信内容。 |

返回值： Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

### 3.3.1.8 sendSM(mobiles, content , smID , srcID , URL)

发送一条 **WAP Push** 短信到手机号码集。

**原型：** public int sendSM(String[] mobiles, String content , long smID , long srcID , String url)

**参数：**

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | WAP Push 短信的标题 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执 |
| srcID | 终端源地址，0 到 99999999 中的某一整数。用来作源地址显示在终端上，确保唯一后可以用来找到对应的回复 |
| url | WAP Push 短信的 URL，为“”或 null 时则发送常规短信，此时 content 为短信内容。 |

**返回值：** Int. IMAPI_SUCC , IMAPI_INIT_ERR IMAPI_CONN_ERR , IMAPI_DATA_ERR , IMAPI_DATA_TOOLONG , IMAPI_IFSTATUS_INVALID , IMAPI_GATEWAY_CONN_ERR

### 3.3.1.9 sendSM(mobiles, content , sendTime , smID , srcID)

定时发送一条 **MT** 短信到一个手机号码集。

**原型：** public int sendSM(String[] mobiles, String content, String sendTime, long smID, long srcID)

参数：

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | 普通短信内容，最大支持 2000 个汉字的短信，超过 2000 时会被截断。 |
| sendTime | 定时发送时间，格式为“yyyy-MM-dd HH-mm-ss”，为 null 时立即发送 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执 |
| srcID | 终端源地址，0 到 99999999 中的某一整数。用来作源地址显示在终端上，确保唯一后可以用来找到对应的回复 |

返回值：Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

### 3.3.1.10 sendSM(mobiles, content , smID , srcID, URL , sendTime)

定时发送一条 MT 短信到一个手机号码集。

原型：public int sendSM(String[] mobiles, String content, long smID, long srcID, String url, String sendTime)

参数：

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | 普通短信内容，最大支持 2000 个汉字的短信，超 |

过 2000 时会被截断。

| | |
|---|---|
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执 |
| srcID | 终端源地址，0 到 99999999 中的某一整数。用来作源地址显示在终端上，确保唯一后可以用来找到对应的回复 |
| url | WAP Push 短信的 URL，为“”或 null 时则发送常规短信。 |
| sendTime | 定时发送时间，格式为“yyyy-MM-dd HH-mm-ss”，为 null 时立即发送 |

返回值：Int. IMAPI_SUCC，IMAPI_INIT_ERR IMAPI_CONN_ERR，IMAPI_DATA_ERR，IMAPI_DATA_TOOLONG，IMAPI_IFSTATUS_INVALID，IMAPI_GATEWAY_CONN_ERR

### 3.3.1.11 sendPDU(mobiles, content, smID, msgFmt, tpPID, tpUdhi, feeTerminalID, feeType, feeCode, feeUserType)

不支持 **srcID** 的方法，发送一条 **PDU** 短信到一个手机号码集。

原型：public int sendPDU(String[] mobiles, byte[] content, long smID, int msgFmt, int tpPID, int tpUdhi, String feeTerminalID, String feeType, String feeCode, int feeUserType)

参数：

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |

| | |
|---|---|
| content | PDU 短信内容，最大支持 140 个字节的短信，超过 140 个字节，发送时会被截断。 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |
| msgFmt | 短信编码格式（最终通过 cmpp2.1 协议发送至网关的短信格式，具体类型请参见 cmpp2.1 或 cmpp3.0 协议文档；）<br>默认为 0。 |
| tpPID | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| tpUdhi | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| feeTerminalID | 被计费用户的号码，默认为空。 |
| feeType | 资费类别，默认为空。 |
| feeCode | 资费代码，默认为空。 |
| feeUserType | 计费用户类型，默认为 0。<br>当值为 -1 时，上面三个计费参数（feeTerminalID、feeType、feeCode）无效，MAS 自动取通讯设置中的计费参数值。 |

**返回值：** Int. IMAPI_SUCC，IMAPI_INIT_ERR IMAPI_CONN_ERR，IMAPI_DATA_ERR，IMAPI_DATA_TOOLONG，IMAPI_IFSTATUS_INVALID，IMAPI_GATEWAY_CONN_ERR

### 3.3.1.12 sendPDU(mobiles, content , smID , srcID, msgFmt, tpPID, tpUdhi, feeTerminalID, feeType, feeCode, feeUserType)

支持 **srcID** 的方法，发送一条 **PDU** 短信到一个手机号码集。

**原型：** public int sendPDU (String[] mobiles, byte[] content, long smID, long srcID, int msgFmt, int tpPID, int tpUdhi, String feeTerminalID, String feeType, String feeCode, int feeUserType)

**参数：**

| | |
|---|---|
| mobiles | 短信发送的多个目的手机号码的数组 |
| content | PDU 短信内容，最大支持 140 个字节的短信，超过 140 个字节，发送时会被截断。 |
| smID | 短信 ID，0 到 99999999 中的某一整数。确保唯一后可以用来找到对应的回执、回复 |
| srcID | 终端源地址，0 到 99999999 中的某一整数。用来作源地址显示在终端上 |
| msgFmt | 短信编码格式（最终通过 cmpp2.1 协议发送至网关的短信格式，具体类型请参见 cmpp2.1 或 cmpp3.0 协议文档；）<br>默认为 0。 |
| tpPID | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| tpUdhi | 默认为 0，具体请参见 cmpp2.1 或 cmpp3.0 协议文档。 |
| feeTerminalID | 被计费用户的号码，默认为空。 |

feeType　　资费类别，默认为空。

feeCode　　资费代码，默认为空。

feeUserType　　计费用户类型，默认为 0。

当值为 -1 时，上面三个计费参数（feeTerminalID、feeType、feeCode）无效，MAS 自动取通讯设置中的计费参数值。

**返回值：** Int. IMAPI_SUCC ， IMAPI_INIT_ERR IMAPI_CONN_ERR ， IMAPI_DATA_ERR ， IMAPI_DATA_TOOLONG ， IMAPI_IFSTATUS_INVALID ， IMAPI_GATEWAY_CONN_ERR

### 3.3.1.13 receiveSM()

接收所有已收到的 MO 短信。

**原型：** public MOItem[] receiveSM ()

**参数：**

无。

**返回值：** MOItem[]。所接收的短信的数组。null 表示接收失败，数组长度为 0 表示无短信。

### 3.3.1.14 receiveSM(amount)

根据指定数量接收收到的 MO 短信。

**原型：** public MOItem[] receiveSM (int amount)

**参数：**

| | |
|---|---|
| amount | 要取得的 MO 短信条数，当其值为-1 时，表示接收所有数量的回复； |

**返回值：** MOItem[]。所接收的短信的数组。null 表示接收失败，数组长度为 0 表示无短信。

### 3.3.1.15 receiveSM(srcID , amount)

根据指定的 **srcID** 和指定数量接收收到的 **MO** 短信。

**原型：** public MOItem[] receiveSM (long srcID, int amount)

**参数：**

| | |
|---|---|
| srcID | 终端源地址，0 到 99999999 中的某一整数。与发送 MT 短信时的 srcID 对应，当其值为-1 时，表示此参数不起作用，只按接收的条数 amount 取 MO 短信。 |
| amount | 要取得的 MO 短信条数，当其值为-1 时，此参数不起作用，只按 srcID 取 MO 短信。当 srcID 也为-1 时，此时此方法等同于无参数的 receiveSM()方法，接收所有 MO 短信。 |

**返回值：** MOItem[]。所接收的短信的数组。null 表示接收失败，数组长度为 0 表示无短信。

### 3.3.1.16 receiveRPT()

接收所有已返回的短信回执。如果短信接口不支持回执，就返回 **null**。

**原型：** public RPTItem[] receiveRPT()

**参数：**

无。

**返回值：** RPTItem[]。所接收的回执的数组。null 表示接收失败，数组长度为 0 表示无回执。

### 3.3.1.17 receiveRPT(amount)

根据接收数量接收已返回的短信回执。如果短信接口不支持回执，就返回 **null**。

**原型：** public RPTItem[] receiveRPT(int amount)

**参数：**

| | |
|---|---|
| amount | 要取得的短信回执的条数。当其值为-1 时，表示接收所有数量的回执； |

**返回值：** RPTItem[]。所接收的回执的数组。null 表示接收失败，数组长度为 0 表示无回执。

### 3.3.1.18 receiveRPT(smID , amount)

根据 **smID** 和接收数量接收已返回的短信回执。如果短信接口不支持回执，就返回 **null**。

**原型：** public RPTItem[] receiveRPT(long smID, int amount)

**参数：**

| | |
|---|---|
| smID | 短信 ID，0 到 99999999 中的某一整数。与发送 MT 短信时的 smID 对应。当其值为-1 时，表示此参数不起作用，只按接收的条数 amount 接收回执。 |
| amount | 要取得的短信回执的条数。当其值为-1 时，此参数不起作用，只按 smID 取短信回执。当 smID 也为-1 时，此时此方法等同于无参数的 receiveRPT()方法，接收所有已返回的短信回执。 |

**返回值：** RPTItem[]。所接收的回执的数组。null 表示接收失败，数组长度为 0 表示无回执。

#### 3.3.1.19 release()

本方法用于断开到移动代理服务器的连接，同时释放相关资源。

**原型：** public void release()

**参数：**

无。

**返回值：** 无。

### 3.3.2 类 MOItem

保存一条 MO 短信的数据类。

**属性**

| 属性 | 属性说明 |
|---|---|
| Private String mobile | 发送短信的手机号码 |
| Private long smID | MO 短信的短信 ID，对应 MT 的 srcID |
| Private String content | 短信内容 |
| Private String moTime | MO 短信时间 |
| Private int msgFmt | MO 短信的编码格式 |

**方法**

无。

### 3.3.3 类 RPTItem

保存一条短信回执的数据类。

**属性**

| 属性 | 属性说明 |
| --- | --- |
| Private String mobile | 发送短信的手机号码 |
| Private long smID | MT 短信的短信 ID |
| Private int code | 回执编码 |
| Private String desc | 回执描述 |
| Private String rptTime | 回执时间 |

**方法**

无。

# 4 使用指导

## 4.1 开发步骤

对于使用移动代理服务器的用户，利用移动代理服务器短信接口进行其它软件的开发与使用，大体分为如下步骤：

**接口创建：** 登录移动代理服务器，进入系统管理->接口管理模块，根据需要创建一个合适的短信接口

**开发：** 根据将使用的开发语言选择对应语言的接口语言包进行开发，使用接口类型为 API 接口。如果没有对应的语言包，就使用 DB 接口。如果使用 DB 接口，则在第 1 步创建短信接口时要创建 DB 类型的接口。

**调试与测试：** 编码过程中或编码完成后，利用接口管理模块的队列管理作为一个短信模拟终端进行调试与测试。

**运行：** 在开发完成后，把对应接口设置为运行状态，就可以上线运行所开发的软件。注意：如果移动代理服务器已经接入运营商的短信网络，那么在把对应接口修改为运行状态之前，请确保已清空发送队列中的所有调试数据，否则在把对应接口修改为运行状态之后，将自动对发送队列中的所有数据进行发送。

## 4.2 调试方法

当利用移动代理服务器提供的短信接口进行软件开发时，直接使用手机进行短信收发的调试当然是可以的。但是如果移动代理服务器还没有接入运营商的短信网络，或者考虑到调试中所产生大量短信费用的节省，或者为了确保开发中不会造成陌生手机用户的骚扰短信投诉事件，这时候可以考虑使用接口管理模块的队列管理作为短信模拟终端进行调试与测试。

要使用接口管理模块的队列管理作为短信模拟终端使用，必须把对应接口的状态设为调试状态。短信接口是否处于调试状态以及把短信接口切换到调试状态的方法请参见第 **3** 节。

当对应接口的状态为调试状态时，先进入接口管理模块的队列管理主页面，然后通

过如下方法进行 MO 短信、MT 短信、MT 短信的回执进行调试：

**MO 短信调试：** 通过在发送队列中新增 MO 短信记录来模拟手机的 MO 短信发送。随后所开发软件就能接收到此 MO 短信。

**MT 短信调试：** 所开发软件发送的 MT 短信，可以在接收对列中查收。

**回执调试：** 通过在回执队列中新增回执记录来模拟来自短信网关的回执。随后所开发软件就能接收到此短信回执。

在调试状态时，队列管理主页中的发送队列、接收队列、回执队列都提供了删除功能，开发人员可以根据需要进行记录的删除与队列清空。

## 4.3 运行设置与监控

只要移动代理服务器通讯连接设置正确，并已正常运行，在短信接口置于运行态下，接入短信接口的外部系统就可正常运行。短信接口是否处于运行状态以及把短信接口切换到运行状态的方法请参见第 **3.1** 节。

MO 短信的分流支持两种方式，要使手机指令分流方式有效，就要确保短信接口所设置的手机指令正确。短信接口手机指令的设计方式请参见第 **3.1** 节。

另外，移动代理服务器中系统管理的短信过滤模块的过滤词组设置、黑名单设置、白名单设置都会对短信接口的短信发送有影响，所以要检查其设置是否合适。过滤词组、黑名单、白名单的设置方法请参见《移动代理服务器 **MAS** 使用手册**.doc**》。

短信接口运行时，可通过接口管理模块进行监控，具体参见第 **3.1** 节。

短信接口所收发的所有短信都有短信话单，这些话单可以通过话单管理模块查询，也可以利用统计报表模块进行报表统计。话单管理、统计报表模块的具体操作方法请参见《移动代理服务器 **MAS** 使用手册**.doc**》。

# 5 附录

## 5.1 附录一：API 返回值列表

| 返回值 | 返回描述 | 属性名称 |
|---|---|---|
| 0 | 操作成功 | IMAPI_SUCC |
| -1 | 连接数据库出错 | IMAPI_CONN_ERR |
| -2 | 数据库关闭失败 | IMAPI_CONN_CLOSE_ERR |
| -3 | 数据库插入错误 | IMAPI_INS_ERR |
| -4 | 数据库删除错误 | IMAPI_DEL_ERR |
| -5 | 数据库查询错误 | IMAPI_QUERY_ERR |
| -6 | 参数错误 | IMAPI_DATA_ERR |
| -7 | API 编码非法 | IMAPI_API_ERR |
| -8 | 参数超长 | IMAPI_DATA_TOOLONG |
| -9 | 没有初始化或初始化失败 | IMAPI_INIT_ERR |
| -10 | API 接口处于暂停（失效）状态 | IMAPI_IFSTATUS_INVALID |
| -11 | 短信网关未连接 | IMAPI_GATEWAY_CONN_ERR |

## 5.2 附录二：回执编码列表

所有短信回执的回执定义如下列表：

| 回执编码 | 回执描述 | 说明 |
|---|---|---|
| 0 | 发送成功 | 手机已收到短信。 |
| 1 | 发送内容为空 | 没有短信内容。 |
| 2 | 发送内容中存在被禁止词组 | 短信内容中包含短信过滤模块中所设置的过滤词组。 |
| 3 | 手机号码不正确 | 不是正确的手机号码。 |
| 4 | 手机号码为运营商所禁止 | 此手机号码的短信发送不为运营商所支持或被运营商所禁止 |
| 5 | 手机号码存在黑名单中 | 此手机号码在短信过滤模块中所设置的黑名单范围中。 |
| 6 | 手机号码不存在白名单中 | 此手机号码不在短信过滤模块中所设置的白名单范围中。 |
| 7 | 企业欠费 | 企业帐户余额已用完。 |
| 8 | 通讯异常 | 与短信网关的连接出现异常，此时不能发送短信 |
| 101 | 系统错误[n] | 系统内部出错。频繁出现此问题请联系开发商。 |
| 102 | 短信内容无法到达手机[n] | 短信中心无法把短信发送到对应手机。可能是手机不存在、手机停机、手机关机、手机接收信号不好、短信中心出问题等原因所造成。 |

地址：深圳市科技园高新南一道创维大厦 A 区 801 号

电话：0755-26743535 传真：0755-26743155 邮编：518057